CLAIMS

1. A method for securing by software confinement, a computer system which executes codes which manipulate data, involving:

- at least one memory manager managing memory allocation units which may typically be a page with a fixed size or a block with a variable size,
- at least possessors and requesters of memory allocation units which may typically be an application of the user of the operating system of the computer system or the operating system itself,

characterized in that it comprises the following steps:

- an allocation of memory performed by the memory manager upon request from another component of the operating system which transmits to said memory manager, the identity of the requester;
- a check by the aforesaid memory manager of the whole of the allocation units, each being associated with a possessor of the memory allocation unit;
- an encryption of the data of each possessor by means of a key associated with this possessor;
- a check by the memory manager, for each request to access a memory allocation unit, of the identity of the requester; if this identity is not identical to that of the possessor of said memory allocation unit, then access to the memory allocation unit is refused by the memory manager;
- performing, by means of the memory manager, encryption (in the case of a write request) or decryption (in the case of a read request) of the relevant data with the key associated with the possessor, this key being at least recalculated by the memory manager.

2. The method according to claim 1, characterized in that the allocation unit is the page, and the memory manager, when it receives a request for allocating a block on behalf of a possessor of a memory allocation

unit, first searches for a page with the same possessor so that all the blocks allocated by said possessor are found grouped in one or several dedicated pages.

3. The method according to claim 1, characterized in that transmission of the identity of the requester is accomplished either by managing a current context, or by passing parameters to the functions of the memory manager.

4. The method according to claim 1, characterized in that the memory manager dynamically calculates the key of a possessor from a secret associated with said possessor and a so-called master key to which only the memory manager has access.

5. The method according to claim 1, characterized in that the memory manager associates the key with each set of possessor and memory allocation unit instead of associating a unique key with each possessor.

6. The method according to claim 1, characterized in that the memory manager integrates into each memory allocation unit, an area with which the integrity of the latter may be checked.

7. The method according to claim 1, characterized in that it associates different security levels with the applications and uses different encryption means according to the associated security level.

8. The method according to claim 1, characterized in that it is combined with a physical protection mechanism.

9. The method according to claim 1, characterized in that it is

implemented on an embedded system such as a terminal of the portable telephone type, a bank payment terminal, a portable payment terminal, a digital assistant or PDA, a chip card.